

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **06-295286**

(43)Date of publication of application : **21.10.1994**

(51)Int.Cl.

**G06F 15/16**

**G06F 15/00**

**H04L 9/00**

**H04L 9/10**

**H04L 9/12**

(21)Application number : **05-079302**

(71)Applicant : **FISCHER ADDISON M**

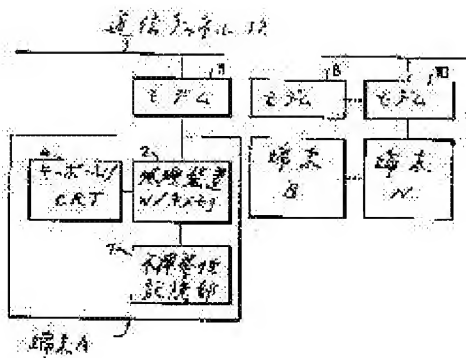
(22)Date of filing : **06.04.1993**

(72)Inventor : **FISCHER ADDISON M**

(30)Priority

Priority number : **92 863552** Priority date : **06.04.1992** Priority country : **US**

(54) **METHOD FOR PROCESSING INFORMATION BETWEEN COMPUTERS OF COMMUNICATION SYSTEM**



(57)Abstract:

PURPOSE: To improve the information processing between computers by giving a specific programming instruction to the computers.

CONSTITUTION: In a communication system, terminals A, B,..., N are connected to a communication channel 12 through MODEMs 610 and each terminal A, B,..., N is constituted of a keyboard/CRT 4, a processor 2 provided with a main memory, and a nonvolatile storage device 7 and can transmit and receive messages containing a 'move program'. The 'move program' is constructed in a digital data structure and has continuous instructions, related data, and an ability for deciding the next destination of the 'move program' and transmits the 'move program' to the destination.

Therefore, each terminal A, B,..., N can generate messages, can load peculiar logic, data, and function on the 'move program', can enable digital signing operations required for execution, and can transmit the messages to the terminals connected to the channel 12.

## LEGAL STATUS

[Date of request for examination] 28.03.2000

[Date of sending the examiner's decision of rejection] 14.12.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection] 2005-004066

[Date of requesting appeal against examiner's decision of rejection] 09.03.2005

[Date of extinction of right]

\* NOTICES \*

**JPO and INPIT are not responsible for any damages caused by the use of this translation.**

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

CLAIMS

---

[Claim(s)]

[Claim 1] In the communication system with which the computer was combined with the exchangeable channel (12) for two or more computers (terminals A, B, --, N) in the message through it Are an approach for processing information between said computers, and the step which gives a series of program instruction ( drawing 2 , block 22) to the 1st computer is included. Program instruction is executed by 1st computer and the instruction which determines the destination of at least one degree which should receive the instruction of a group is included. The instruction of said group includes the instruction for transmitting said instruction to said following destination with attached data. The contents are the approaches containing the step which performs a digital signature (432) based on said digital value at at least one destination further based on the logical decision and processing which are performed by said program including the step which calculates digital value.

[Claim 2] Said digital signature is an approach according to claim 1 expressed as data exposed to being logically processed by said a series of program instruction.

[Claim 3] The digital certification relevant to said digital signature is an approach according to claim 1 expressed as data exposed to being logically processed by said a series of program instruction.

[Claim 4] Said digital signature is an approach according to claim 1 included as said some of attachment data transmitted to the following destination.

[Claim 5] The step converted into the specialized DS which follows partially at least the criterion which had said data recognized by said a series of program instruction is included further, and said DS is a useful approach according to claim 1 regardless of said instruction by it.

[Claim 6] The approach according to claim 5 of containing further the step which processes and corroborates the data with which a digital signature and it are given regardless of said program instruction.

[Claim 7] Data are an approach according to claim 1 converted into the electronic data exchange (EDI)

format standardized under directions of a series of instructions.

[Claim 8] This information is an approach according to claim 1 treated as a program variable to which a series of program instruction operates based on it including the step to which a digital signature assembles logically the information which may be given alternatively.

[Claim 9] A digital signature is an approach according to claim 1 performed as a function which may be called under control of said a series of program instruction.

[Claim 10] The data supplied to said digital signature function are the approach according to claim 9 of reflecting a value based on the group of the arbitration of the data read from a user's file, the data incorporated in said a series of program instruction, the data inputted by the user, the data obtained from other signatures, and the data obtained from digital certification (514).

[Claim 11] The approach according to claim 1 of including further the display of the authority granted to the user who performs a digital signature (432) by including sufficient digital information for the authority which signature people used enabling the corroboration of what (434) it used appropriately.

[Claim 12] The approach according to claim 1 of containing further the step which chooses the certification which should be used in case a digital signature (434) is performed from collection of digital certification.

[Claim 13] In the communication system combined with the channel (12) with a computer able for two or more computers (terminals a, b, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives a series of instructions ( drawing 2 , block 22) to the 1st computer is included. An instruction is executed by 1st computer and the instruction which determines the destination of at least one degree which should receive the instruction of a group is included. The step which gains data from at least one user of said computer by activation of said instruction including an instruction for the instruction of said group to transmit said instruction to said following destination with attached data, The step converted into the specialized DS which follows at least the criterion which had said data recognized partially through activation of said instruction is included. It is an approach containing the step which is useful regardless of said instruction as for said DS, and signs said DS in digital one through activation of said instruction further by it.

[Claim 14] Data are an approach according to claim 13 converted into the electronic data exchange (EDI) format standardized under directions of an instruction of a group.

[Claim 15] The approach containing the step converted by using an EDI translator according to claim 13.

[Claim 16] An EDI translator is the approach according to claim 15 of being the external module called under control of said instruction.

[Claim 17] Some aggregates [ at least ] of said DS are the approaches according to claim 13 transmitted as a group of data regardless of the instruction of a group with the digital signature of said

DS.

[Claim 18] Some aggregates [ at least ] of said DS are the approaches according to claim 13 stored separately from the instruction of a group with the digital signature of said DS.

[Claim 19] The result of a digital signature is an approach according to claim 13 stored as some attachment data.

[Claim 20] The result of a digital signature is an approach according to claim 13 corroborated when the instruction of a group is executed at the destination of at least one consecutiveness.

[Claim 21] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives the 1st migration program including a series of instructions which determine the destination of at least one degree which should receive the instruction of a group to a computer is included. The step which gives the 2nd migration program ( drawing 3 7:694 ) at least to one of said computers including an instruction for the instruction of said group to transmit said instruction to said following destination with attachment data, The approach containing the step which performs the 2nd migration program under directions of the 1st migration program.

[Claim 22] The 1st and 2nd migration programs are the approaches according to claim 21 of being the instance from which the instruction of the same group differed.

[Claim 23] The 1st and 2nd migration programs are approaches including the instruction of the group which is completely different according to claim 21.

[Claim 24] The 1st migration program is a method according to claim 21 of giving data to the 2nd migration program which specifies the actuation which should be performed by the 2nd ( drawing 3 7:694 ).

[Claim 25] The 2nd migration program is a method according to claim 21 of returning data to the 1st migration program ( drawing 3 7:687 and 706).

[Claim 26] The approach according to claim 21 by which both migration programs are transmitted in a high-level interpretation format so that a migration program and data may be interpreted based on various computer systems and hardware architecture.

[Claim 27] An interpretation format is the approach according to claim 26 which can be processed by computer of at least two molds which are completely different.

[Claim 28] The 1st migration program is the approach according to claim 21 of eliminating the 2nd migration program from memory.

[Claim 29] The 2nd program instance is an approach according to claim 21 saved after the activation.

[Claim 30] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives the 1st migration program

instance which includes a series of instructions ( drawing 2 , block 22) in a computer is included. An instruction is executed by computer and the instruction which determines the destination of at least one degree which should receive the instruction of a group is included. The step which gives said migration program instance ( drawing 3 7:694 ) to at least one of said the computers including an instruction for the instruction of said group to transmit said instruction to said following destination with attachment data, The approach containing the step which processes the 2nd migration program under directions of an instruction of the 1st migration program instance.

[Claim 31] Processing actuation is an approach containing the step which eliminates the 2nd migration program instance according to claim 30.

[Claim 32] Processing actuation is an approach containing the step which extracts data from the 2nd migration program instance (687 706) according to claim 30.

[Claim 33] Processing actuation is an approach containing the step which changes the program instruction of the 2nd migration program instance according to claim 30.

[Claim 34] Processing actuation is an approach containing the step which changes the value of the variable stored in the 2nd migration program instance according to claim 30.

[Claim 35] Said 2nd program instance is an approach including the same instruction as the 1st program instance according to claim 30.

[Claim 36] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives a series of instructions ( drawing 2 , block 22) to the 1st computer is included. An instruction is executed by the 1st computer and the instruction which determines the destination of at least one degree which should receive the instruction of a group is included. The step which the instruction of said group answers activation of said the instructions of a series of further including the instruction for transmitting said instruction to said following destination with attachment data, and chooses a file, The approach containing the step which answers said a series of instructions ( drawing 35 , drawing 36 , drawing 37 , 640-642, 680-708, 710-732), and transmits some contents [ at least ] of said selected data file to said following destination.

[Claim 37] The approach containing the step which signs in digital one in some data [ at least ] of said file according to claim 36.

[Claim 38] The approach containing the step which calculates some [ at least ] hash values of the data of said file according to claim 36.

[Claim 39] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it It is an approach for sending information with said communication system. The step which gives 1 set of instructions ( drawing 2 , block 22) to the 1st computer is included. It performs by 1st computer including the instruction which the instruction of the group generates two or more instances of an instruction of said

group, and receives one of said instances with attachment data, respectively and which starts transmission to the 1st and 2nd destinations at least. The approach containing a step including the capacity by which the data stored between the transmitting path ( drawing 37 ) which is completely different into said instance furthermore transmitted to said 1st and 2nd destinations are merged later.

[Claim 40] The approach according to claim 39 of containing further the step which establishes one instance as a master instance (682), and the step which extracts data from other instances when a master is controlled further and other instances reach a merge destination.

[Claim 41] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives a series of program instruction ( drawing 2 , block 22) to the 1st computer is included. The program instruction is executed by 1st computer, and the instruction which determines the destination of at least one degree which should receive the instruction of the group is included. The instruction of said group is an approach containing the step which embellishes actuation of the group which has it permitted that said a series of instructions execute said instruction further including the instruction for transmitting to said following destination with attachment data.

[Claim 42] Said qualification means is an approach according to claim 41 specified by the person concerned who uses said program.

[Claim 43] Said qualification means is an approach according to claim 41 signed in digital one by one person concerned trusted by the person concerned who uses said migration program.

[Claim 44] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives a series of program instruction ( drawing 2 , block 22) to the 1st computer is included. The program instruction of a series of is executed by 1st computer. The instruction which determines the destination of at least one degree which should receive the instruction of the group is included. The instruction of said group is an approach containing the step which performs a digital signature by using the individual key further stored in the user token device including the instruction for transmitting said instruction to said following destination with attachment data.

[Claim 45] The approach containing the step which determines whether user's individual's key is stored in the token device or the computer memory, and the step which is said token device or processes a signature by a user's computer, respectively according to claim 44.

[Claim 46] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals A, B, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives a series of program instruction ( drawing 2 , block 22) to the 1st computer is included. The program instruction of a series

of is executed by 1st computer. The instruction of said group is an approach containing the step which performs a date / time amount authentication further including the instruction for transmitting said instruction to said following destination with attachment data including the instruction which determines the destination of at least one degree which should receive the instruction of the group.

[Claim 47] In the communication system combined with a channel (12) with a computer able for two or more computers (terminals a, b, --, N) to exchange messages through it Are an approach for processing information between said computers, and the step which gives a series of program instruction ( drawing 1 , 22) to the 1st computer is included. The program instruction is executed by 1st computer, and the instruction which determines the destination of at least one degree which should receive the instruction of the group is included. The instruction of said group is an approach containing the step which performs a time amount delay function ( drawing 9 : 570) further including the instruction for transmitting said instruction to said following destination with attachment data.

---

[Translation done.]

\* NOTICES \*

**JPO and INPIT are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

#### DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001] This invention is "migration" (travelling) which has the capacity to move itself to other computer users with required associated data from one computer user, and serves as a powerful tool for data processing in various computer nodes, data authentication, and data collection by that cause. It is related with the approach and equipment for creating a program.

[0002]

[The background and epitome] of invention In an in-house, a document often moves through a help. When a sentence document needs to be exchanged among two or more organizations, mail or delivery service is often adopted.

[0003] About the in-house and the electronic document transmission technique during an organization, it is known well. A commercial transaction of a certain kind was automated by rapid development of a



electronic-mail system, an electron transport system, etc., and when it was many, some of transfer of the document by the unnecessary help was lost.

[0004] The so-called "electronic formal" methodology is used for the methodology of a certain advanced technology for transmitting information automatically among users (for example, in-house). It corroborates this "electronic formal" methodology having offered data to the user, having urged a user's input to it with the indicating equipment of the conventional technique, and having been correctly put into the inputted data, and such [ after that ] data are transmitted to other users.

[0005] In many viewpoints, the methodology of this electronic format is limited very much. For example, when data express a certain value, actuation, an alteration, or the danger of only being forged always hangs around [ data ]. Attaching a flag to the important field of a certain kind which should be signed in digital one was included in the attempt coping with this danger. Thereby, about specific input field, although it is a certain limited amount, the authentication made completely the same as that into which it was put is attained.

[0006] However, by this approach, it is impossible in complicated DS an assembly and to sign in digital one after that. No matter what algorithm this application may follow, it makes it possible to calculate the digital data which should be corroborated if needed [ that a migration program should be signed / the digital data and if needed ].

[0007] In this way, for example, according to the invention in this application, the actual data signed become possible [ coming to differ from any field data itself ]. The signed data are able not to actually contain at all the actual data offered by the user, either.

[0008] For example, one mode especially with this useful is the case where an EDI (electronic data exchange) transaction is created based on the aspect of affairs of data where the migration program of the invention in this application was entered. A program has the capacity which signs an EDI transaction. Such an EDI transaction may consist of complicated digital information by which the lookup was carried out based on the supervisor who drives the file or this migration program of other tables, or the internal table within the program from an interpreter. In this way, the actual digital data signed with the input field possibly only entered as "X" chosen from a certain kind of table completely differ.

[0009] It is thought that the type of the above-mentioned digital signature may be corroborated by what can consider the application to a data configuration with a long life, and is probably different through a \*\*\*\*\* period. For example, in the case of EDI, a signature can be connected to the EDI transaction itself, it may be corroborated by the future sink of the transaction of a parenthesis, and what separated from the context of a migration program is sufficient as it. This type of digital signature is similar to the signature by the handwriting in the part under the purchase order sheet of paper, or a contract.

[0010] According to [ in addition to the point which can sign the data of arbitration ] the invention in

this application, it is possible in a program determining conditional which user should participate in this signature process based on a certain known criteria.

[0011] For example, according to the invention in this application, this migration program can make a decision logical within the limits of a program about what kind of joint signature the need for may exist about data, a specific user, or specific combination. The information included in the digital certification (for example, thing to depend on invention-in-this-application people's U.S. Pat. No. 4,868,877 or No. 5,005,200) with which the user was reinforced [ were reinforced and it was X.500(certificate )-proved ] may be included in this. Since the flexibility as a perfect program exists, it is even possible to use the information taken out in this way for adjustment of the future transmission root of this migration program.

[0012] In addition to the point which uses a digital signature for easy authentication, the invention in this application makes possible what is corroborated including the requirements for authorization, and use. This uses the No. 4,868,877 and No. [ 5,005,200 ] instruction of managing the certification and a delegation of authorization.

[0013] On the other hand, the invention in this application enables a migration program to offer precious authentication of other types again using a digital signature. For example, it becomes possible to perform digital signature authentication about all the transmissions to other users [ user / one ] as security by the invention in this application being expedient. This includes the migration program itself, its variable and auxiliary data, or a file.

[0014] The authentication based on [ above-mentioned at the point that this 2nd type of digital authentication holds a prolonged signature partially ] data is because the variable and other data which are different, that is, are transmitted are changed when the user by the side of a sink takes action once. Since it goes back and the injustice which is not permitted even if it follows, and it thinks as an inaccurate preventive measure fundamentally and it is based on one person of the real user of this format again was inspected in the past legally, this 2nd type of authentication may be used.

[0015] In addition, the invention in this application provides authentication of the 3rd type with which sign, authentication, and authorization may be carried out to the migration program itself by the right person of issue (for example, author) who can set a certain reliance, in order to check that what kind of bug or a "virus" has not invaded again (this prevents even infection by the user who has the effective ownership of the program along the root).

[0016] The invention in this application offers the original device for automation of the data collection between a user's groups. A migration program is sent to one user, and a related data file is attached (or separation), and it can move to the next user. The data or the file collected from one person or two users or more is kept by other users, or if it is a request, it may be accumulated for batch processing. It becomes unnecessary to depend on each user transmitting all required data in a required format according to this methodology.

[0017] This invention performs effectively the electronic document interchange (EDI) in the context of a migration program by which itself is sent to the next user from the user in an in-house again, collection of data, edit, and approval. When [ suitable ] determined by the logic of a program, in order to transmit to other organizations, it is possible to generate a standard EDI transaction (for example, X12 850 purchase order transaction set) in a program. A migration program can sign the set of the ended transaction in digital one. Therefore, if it is an organization by the side of the sink who can process standardized EDI and the standardized signature, the organization by the side of the sink will become possible [ attesting and processing the data which are taught by the invention in this application and which carry out Iriki ], even if it does not have all available powerful techniques.

[0018] On the contrary, it is enabled for a migration program to receive a general EDI transaction and the thing by which the \*\* sign was probably carried out, and for the syntax of them to be analyzed by this invention, and to be included in that variable. A migration program corroborates an input there and has the capacity to move between various sinks a nest and if needed into a display.

[0019] It will reach, and other descriptions will be understood better by reading explanation of the desirable example of the following invention, when [ of the invention in this application / these ] seeing with an attached drawing.

[0020]

[Detailed explanation of a desirable example] Drawing 1 is the block diagram showing usable instantiation-communication system about the invention in this application. This system contains the communication channel 12 with which Terminals A, B, --, N can communicate on it. The communication channel by which security processing of the telephone line etc. is not made is sufficient as a communication channel 12.

[0021] Terminals A, B, --, N are IBM which has the processor (main memory) combined with the keyboard / CRT display 4 of the conventional technique by only the instantiation purpose. PC compatibility computer may be used. The processor which has main memory 2 is combined also with nonvolatile storages, such as disk memory. Each terminals A, B, --, N are IBM of the conventional technique which enables a terminal to transmit and receive a message including a migration program, when combined with the modem (respectively 6, 8, 10) of the conventional technique again. PC communication board (not shown) is included.

[0022] the "migration program" used by this detail letter is digital data structure, and it has the capacity to determine the destination or sink of at least one degree for receiving a migration program and transmitting itself to a following sink or a following destination with all the associated data determined by the program, including a continuous instruction and associated data.

[0023] Each terminal can generate a message, although the logic of a proper, data, and a function are loaded and performed to the migration program, whether being unnecessary and the becoming digital signature actuation can also be attained (this is explained more to a detail within this specification),

and the message can be transmitted to other terminals connected to the communication channel 12 (or communication network which may be connected to a communication channel 12 (not shown)).

[0024] The digital signature and certification methodology which are indicated by an artificer's U.S. Pat. No. 4,868,877, No. 5,005,200, and No. 5,001,752 may be used here, and it uses here by citation about these patents. The digital signature methodology of the conventional technique may be used more in alternative.

[0025] Before discussing more the "migration program" structure and the methodology according to the instantiation-example of the invention in this application in a detail, the example of the general actuation in the context of an actual commercial transaction is explained briefly. First, the user of the terminal A of drawing 1 assumes it as the engineer with a comparatively low status who is the member of the design team in the firm which is going to receive component components, in order to complete a circuit design project.

[0026] Suppose that the components demand "a migration program" of a type which this engineer that uses a keyboard 4 explains below at a detail was accessed. A demand "a migration program" urges him to explain the component components needed to an engineer. A migration program is in the higher status within this organization about itself by having the following destination, for example, access to Terminal B, automatically, and this demand is accepted and it contains the instruction sequence automatically transmitted to the senior official who has the authority to sign this in digital one. A migration program may transmit the file which is the need, information, for example, a future destination, auxiliary again, and which may stand on \*\*\*\*\* again. A senior official is urged to sign in digital one suitable for this demand. A digital signature is able to reflect not only a specific adjustable value but an adjustable name. A certain collective structure produced from the variable calculated within a program in alternative may be reflected, and the value may be based on either of many sources containing the data (for example, a user's ID) taken out from the certification of the data built in in the input by the data and the user who were read from the file, and the program, and various signers, or a user's environment.

[0027] If a demand is accepted, the format of a demand will take the path different from the case where this is not accepted, in an in-house. It may have the intelligence which determines where a migration program should transmit itself by the in-house based on the input from the senior official in the terminal B of operation. One from the terminal B which loads the suitable data relevant to a demand to the memory relevant to Terminal B if a migration program is a request again, and needs to be advanced to the somewhere point of an in-house if it is a request of files will be attached.

[0028] Once a signature is made, a migration program has after that the capacity to re-calculate so that data may be corroborated as opposed to every subsequent user, even if it is which reason, and to perform a digital signature corroboration always.

[0029] The result of such a corroboration can be notified also to which sink, or in being many, if a

migration program only corroborates and (it suggests that the alteration of data was performed) has a defect, it will notify of a problem.

[0030] Since a migration program monitor can carry out No. 4,868,877 and No. [ 5,005,200 ] instruction, checking also about authorization is possible so that any sink can check that required authorization has been performed.

[0031] After specific DS is built and a sign under migration program control is performed, it is possible to reconstruct this DS succeeding and to give that signature to other things. Such data must have been altered by neither of the things after this.

[0032] However, this invention realizes capacity to be signed in digital one, in case all the transmitted data are sent to the next user from one user again. The migration program manipulation equipment in a sink's computer can corroborate this signature automatically, if a migration program is loaded. It ensures a component etc. being changed on the way by this, or not being altered. Although only this overall signature expresses the condition of the data between transmissions of this specification and does not have any semantics for a subsequent user, this offers a legal audit device, when the alteration by the participating user who owns this format needs to be pursued, while securing perfect transmission with which it is not interfered by the third person. As for this overall signature, this signature differs from the current capacity for an electronic mail to be signed, in the point that it may be induced by the migration program itself conditional as a part of transmission process.

[0033] Finally, after all authorizations are obtained within organization, a migration process will make an actual purchase order.

[0034] Do this in many modes. A migration program may support some approaches and may choose one most suitable thing according to a given situation. Here, four possibility is described.

[0035] 1. The last purchase order can only be printed out on the form considered that a migration program is mailed physically (it is even possible to print the LOGO of a firm, a letterhead, etc.).

[0036] 2. A migration program can generate automatically the purchase order image considered to appear on a vender's fax machine, if combined with the computer-facsimile capacity outside towards. A purchaser does not need to generate a form.

[0037] 3. If it turns out that a vender supports the migration programming methodology of the invention in this application again, a migration program is able to only specify a vender as the following destination.

[0038] 4. Possibility of saying that it cannot judge clearly whether there being any capacity for a vender to deal with [ a vender / a purchaser's migration program ] the methodology of this migration program, not using the invention in this application is also high.

[0039] Therefore, a migration program operates the in-house data, builds a standard EDI (electronic data exchange) transaction, and is recognized and processed in the range where this is large. A migration program may carry out a digital signature to a computer EDI transaction again, and the both

sides of a signature and a transaction may be transmitted. A migration program transmits an EDI transaction and a certain possible signature to a sink after that (the migration program from a user to the oriented user as a part of migration and its attached transmission of a thing have been independent, and such transmission should not be confused with this).

[0040] If it is the sink who can deal with a standard EDI transaction, the EDI input received also by whom can be treated. If it is the sink who can deal with a digital signature, this transaction can be attested also by whom. Furthermore, when the sink has the software capacity which is sufficient for recognizing these, this sink can corroborate automatically the authentication which may be carried out as a part of signature. It depends on the logic [ whether it is transmitted with the transaction with which certification was signed to which range ] of a migration program.

[0041] In the case of which [ above ], either of the automation of some possible level is used, and a migration program can spin off the purchase order (P. O.) information to a vender. this -- continuing -- a migration program -- one version of itself -- or -- probably -- \*\*\*\*\* -- an alphabetic character is returned to a generating person and it notifies of P.O. having been sent. Other information may be sent to queuing, in order to wait for an archive or the further processing. This information is the record added to the easy message and the file, otherwise, probably a \*\*\*\*\* program plans the traversal (automatic "mailing" or "transmission") of full.

[0042] Drawing 2 shows the migration program according to the instantiation-example of the invention in this application, and its related structure of a component. The migration program of drawing 2 contains the segment of the following multiplex fields at least. Preferably the 1st header segment 20 Each magnitude of a component segment, The name of a related program (and other segments which will be described below if it can do), The date, the type of each component (or a source language program [ For example, a program ], P code by which the program was already compiled), The version of an interpreter required in order to perform discernment of a format, and it, Data (for example, an activation stack, PCB, etc.) with required resuming activation on the suitable point of resumption of a program, the date relevant to the latest traversal, and program authorization information (PAI) are identified most. Each segment in a migration program structure may include an expression of itself so that the type of each component and "size" field "S" may not be included in a header segment 20. According to the purpose of this application, it may be considered that program authorization information (PAI) is the security information as which that a related program carries out specifies the range of the actuation permitted (for example, a definition access to a file). Program call capacity, electronic-mail generating capacity, the capacity that ships data to other users, The capacity which releases a document, the capacity to perform a machine language program, the capacity which accesses the specific field of memory, It is the capacity which accesses the capacity which displays information to a user, the capacity to require a digital signature, digital authentication equipment (digital notary public), etc. About the further detail about the property of program authorization

information, and use, application (surrogate incident number 264-29) of a name called "the computer system method for using program authorization information and equipment" which are application of an applicant for this patent sees. A header segment 20 contains the version number of the migration program of relation again.

[0043] Migration program structure 22 segment is close to the external operation programming language (for example, REXX language), PASCAL, or COBOL preferably reconstructed according to the header of an example, and is written. The thing about the application for example, relevant to a purchase order may be used for the program itself.

[0044] A migration program will have the description which was stated to the above including the capacity to send itself to the further sink. Therefore, a program 22 will include the instruction which sends itself to one person or two sinks or more previously via a certain available medium, and this is known as "traversal" in this specification. It may be the need that one source code instruction or some P-code instructions consider a migration program as a result, and carry out "traversal" to one person or two sinks or more who were identified. The migration program structure shown in drawing 2 is designed so that it may become independent of any specific computer architectures, and it is constituted according to international standards (for example, X.209 format).

[0045] A migration program contains "variable segment" 24 again. Before the 1st user performs, empty is sufficient as most variable segments 24. When it will be specified as the further variable will be required by the program, once a program is sent to a sink, and a program is performed further by that cause, the number of variables will increase. Only for the purpose of instantiation, the variable segment 24 can identify with the actual data value of this variable, a certain variable, for example, "total . dollar . receipt."

[0046] Each variable may have the related information shown in each of the field 32-42 shown in drawing 2 . The field 32 identifies the magnitude of a variable name. The variable name itself is memorized in the field 34. The magnitude of the value of a variable is shown in the field 36. The value of a variable is in the field 38. The field 40 identifies the activation stack level to which a variable belongs. it is because that an activation stack level is identified may exist in the level from which it differs within a program of the same variable name (for example, the variable name same [ that one variable name may exist in the 1st subroutine ] -- an exception -- or it may have a definition which exists in the subroutine which gathered and is moreover different). The activation stack level is useful in order to make it take the same logical structure as what reconstructed the migration program in a sink's computer and it had within an informer's computer. The field 42 is the optional field which can identify the type of a variable, for example, a string, an octet-bit byte, an integer, etc.

[0047] The "variable" segment 24 may contain the digital signature of each variable and related information again. In this way, it becomes possible to reflect the digital signature taken when one or two variables or more were [ between migration program execution paths ] various. One of the

important aspects of affairs of the invention in this application is a point with a migration program able to make a digital signature also to what kind of type of information. This signature itself is carried as a variable. in order to corroborate this signature, it is necessary to show the exact value with which the program was signed (or -- if it to be able to do -- re-calculating), to pass it with a signature value (for this to also be expressed with a variable) after that, and to send to the corroboration signature (VERIFYSIGNATURE) function of a migration program. the digital signature of a variable is included -- a sink side -- data -- one injustice -- not winning popularity -- 2 -- it becomes possible to corroborate that the person who received and did 3 signs of the effective sign obtained authorization appropriately. Please refer to above-mentioned U.S. Pat. No. 5,005,200 which indicates the desirable device for relating authorization with a digital signature.

[0048] For example, the segment 26 of drawing 2 is shown in a migration program including the certification relevant to a certain digital signature so that it may be corroborated by the sink by any signatures at above-mentioned U.S. Pat. No. 5,005,200 as a publication. Certification may be included into a "variable" section with a digital signature in alternative.

[0049] Segment 28A-28N, the file image which it was recorded [ image ] that it becomes possible to attach and memorize the file in which a migration program belongs to the user of a migration program, and had named is included. Then, a user's file may be transmitted with a user's file which others precede with a migration program. It becomes possible to identify the file kept under the situation of specification [ the file or the specific user by whom access of the future of the file by the user becomes easy by naming a file, and the user of a migration program is transmitted to a pan ].

[0050] Including "variable segment" 30, a migration program contains the digital signature of the whole migration structure again so that it may corroborate that this has not received injustice after a sink is seen off for transmission of the whole migration structure at the end. Although migration program DS has been indicated, the related software for performing the DS and the migration program which are used between migration program executions here is explained. Execution control field (XCA) DS is shown in drawing 3 . If a migration program is once received by the sink and XCA is compiled by the P code (it removes when sent by the P code from the beginning), it specifies the information demanded by the program which performs a migration program.

[0051] The XCA segment 82 identifies the address and magnitude of a program which appeared in the file of Iriki as shown in drawing 3 . Please let this whole specification pass, and when a certain segment is described as what memorizes the "address" or a "location", recognize that data are physical or the point that it is the logical address and it is not necessary to necessarily pinpoint an actual physical memory location directly. A program is received by the source or the P code, and the display about which it is is maintained. An execution control field contains the segment 84 showing the address and magnitude of the P-code version of a program. The address (or pointer to the address) of current program control block is identified within a segment 86. For example, the location from which



the list of file control blocks (FCB) used is attached in the file relevant to a migration program, and is removed is shown in a segment 88. The address of the certification regulatory region (CCA) used in order to control the certification attached in a migration program is shown in a segment 90. The location of a "variable" information table (VIT) is shown in the segment 92 which controls and maintains a variable in the form of "B-tree", and this tree is a hierarchy binary tree structure which identifies the location of each program "a variable."

[0052] An execution control field contains the security information segment 94 which may be used in order to corroborate the authentication and authorization which exist in a migration program again. A segment 96 specifies the name of a file including the migration program of Iriki which may need access. A segment 98 pursues the count for which the program itself was mailed in accordance with the Iriki path. The parameter relevant to [ including the input-parameter section 110 again / by this ] program execution in an execution control field may be identified. The execution control field segment 102 identifies the input BEDDA information thought that header information becomes available from the migration program file.

[0053] Drawing 4 shows the DS of the file control block (FCB) used when a migration program attaches or removes a file to itself. or [ that a file control block is attached within a specific user's system ] -- or the tag field 116 which identifies the tag which points out the specific file removed is included. The segment 110 whose file control block is also a pointer to the following file control block is included. A file control block whether it is what was exactly attached by the migration program from which the file related again was received [ whether it may be removed in case it is the time (for example, the following mailing) of a file being the following traversal, and ] The condition segment 112 which the file came outside, or (that is, was the file image of relation loaded in the separate user file?) specifies various conditions of how, a file -- flowing -- a main (stream oriented) thing -- or it has an indicator about "the type of a file" of the thing based on records (record oriented) etc. Other attributes of a file may be specified in this field.

[0054] A segment 114 memorizes the display about the location of the file in the migration program file of main Iriki so that the specific file in question may be accessed quickly. A segment 118 identifies the local name (namely, file name of a migration program most identified by the latest sink) of a file. or [ that a file is attached and the local name of a file is sent to the further sink ] -- or when the already attached file is locally memorized by "it is taken out outside, i.e., a specific user," it is given typically. In addition, FCB may also include the hash of a related file as shown in drawing 4 . If the data used as a foundation can be given, a hash will be "1 Direction" function in which count is easy \*\*\*\*, by computer, as understood by this contractor. If a hash value can be given, a hash function will be \*\*\*\* which cannot make [ also determining the data used as a foundation or ] what kind of data which have the value specified as the hash calculative, either. The value acquired from giving a hash function to the original data aggregate object about all the practical use purposes is the original finger print which

cannot forge the original data. When the original data are changed in one of modes, the hashes of the corrected data differ.

[0055] Drawing 5 shows instantiation-program control block which may be used between migration program executions. Program control block continues a trace of the control information about the program by which one routine is performed in the call and the configuration-ized programming context in which each routine has program control block of relation in other routines.

[0056] The program-control-block segment 50 points to program control block in a program execution control stack to precede. A segment 54 specifies the last performed type of P-code actuation including the segment 52 which specifies the P-code location of a degree where program control block should be performed in a current executive program. A segment 56 contains the pointer to the expression evaluation stack used between expression evaluations. Programme stacks differ typically in that an activation stack is used for a trace of evaluation of an expression of an activation stack, and an internal state. A segment 58 specifies the level of this stack program, and a segment 60 specifies the pointer to the list of variables shared. In REXX language, it may be used in order to access the variable with which the statement "exposed" was shared.

[0057] Drawing 6 shows the adjustable control block data structure (VCB) used in order to control a variable. It identifies whether a variable is located among B trees where, and a segment 62 may contain some pointers. A segment 64 identifies the magnitude of a variable value, and identifies the pointer which, as for a segment 66, the value shows where [ of memory ] it is located. A segment 68 may be optionally used, in order to identify the type of a variable. The variable which shows one of the locations where the segment 70 related to the program after identifying to which level of a migration program the variable would relate and performing the program in it may be deleted easily. Segments 76 and 80 identify each magnitude of a name and name of a variable.

[0058] Here, it returns to explanation of migration program execution. The sequence of the actuation performed by the "loader" section of an interpreter activation drive program is shown in drawing 7 -12. These actuation relates to the preparation for performing a migration program.

[0059] A migration program can be performed in one of the modes in which plurality, such as a batch mode of operation sent to a node, differs, from a node, collecting an interactive user mode, i.e., the mode called by other programs, or information. Initialization information is inputted between the start-up actuation (120) which identifies a specific mode of operation and a specific related execution-time parameter.

[0060] The flow chart shown in drawing 7 -12 shows how the migration program structure shown in drawing 2 is loaded. In case a migration program is loaded, an interpreter makes the execution control field XCA and initial-program control-block PCB. Access to an input parameter is excluded by this, and the name of the input file given since a variable information table (VIT) was loaded and initialized is excluded (122). In the flow chart block 122, an execution control field and program control block

relevant to a migration program are set up. It is filled between the processings which various XCA(s) and PCB fields follow.

[0061] Then, a loader starts loading of the segment of a migration program as shown in drawing 2 , i.e., a header, a program, a variable, certification, a file, and a closure segment. For example, it will fill up with data suitable as follows by loading each of the above-mentioned migration program segments, such as a header program.

[0062] In block 124, decision whether it is necessary to process more segments is performed. If required, an initial input will be read about the segment, and the type of a segment will be judged, and processing of a segment will be started according to the type of a segment after that (126).

[0063] If it returns to header processing of drawing 8 , a check for the segment currently processed to judge whether it is the 1st segment will be performed (150). Since a header must be the first segment when that is not right, an error situation will exist (152). A header is read and hashed when the 1st segment is processed. The data of a header are memorized in XCA (154).

[0064] A routine branches by the entry point L here, and returns to drawing 7 . It judges whether a loader still has the segment which should be processed (124). If it becomes so, processing of a "program" segment as block 126 performed and shown in drawing 9 will be performed as a result. The check which judges whether a header exists or the program is loaded yet at first is performed (160). If an answer is a no, an error situation will exist (162). When an answer is yes, a program is read and a hash is taken (164).

[0065] Then, the digital signature (and/or, header) relevant to a program hash and/or a program is corroborated (166). When a digital signature is not permitted proper or is not corroborated, an error situation arises (166). Generating of a corroboration stores the security and authorization information about a migration program (170). Such authorization information may be held in a header or a program segment in alternative.

[0066] The check which judges whether the program was sent as a P code in the block 172 is performed. When not a P code but a source code is sent, a source code is compiled by the P code using the compiler technique of the conventional technique of common knowledge to this contractor, and a source code image is deleted from storage 174. Location in [ irrespective of / the check in block 172 ] the Iriki file of a program -- -- is stored in XCA irrespective of whether it is the source or it is a P-code format. Copying a program to the traversal which goes to final outside by getting to know the location and range of the Iriki image is simplified. Irrespective of whether finally it was read from whether the P code was only compiled and the Iriki file, the routine with which the primary-storage address and magnitude of a P code are set up in the execution control field (XCA) of 178, and are indicated to be to drawing 7 after that is again entered by block 124, and loading of the segment which remained [ processing / which this shows to drawing 10 / "variable" segment ] arises.

[0067] In processing of a "variable" segment as shown in block 190, the check which determines

whether the variable preceded although the header and the program are loaded has anything is performed. When that is not right, error condition is set to 192 as a result. If anything does not have the variable preceded although loaded as for a header and a program, and it is, the repetitive operation which reads all variables will be started. The check which determines whether there is any variable which should be read by 194 (further) is performed. If there is a variable which should be read further, it will be created as variable control block (VCB) is shown in drawing 6 about each variable, and will be completed with insertion of the variable identifier to variable control block (VCB), and a value, and the set of some condition conditions in VCB. Furthermore, variable control block is added to the suitable spot of a variable information table (VIT), and the table contains all program variables (196). [0068] Furthermore, other variable information about the activation before a migration program is appropriately loaded in a memory stack or program control block in 198, for example. It may be desirable to hold such "control" information to a header segment in alternative here. Then, the check which determines whether a routine needs to branch to block 194 and to read the further variable return and there is performed. This processing is continued until it becomes unnecessary to read a variable any longer, a routine branches to the block 124 of drawing 7 at the point whose need was lost, and the following segment is loaded as a result of [ its ] return.

[0069] As shown in drawing 11 , each certification is read (200), a certification element is created, and it is added to the certification control area (CCA) of storage (202). This process is repeated until all certification is received, as roughly shown in drawing 11 , a routine branches to block 124 at that time, it returns, and the further segment is inspected.

[0070] Probably, it will be desirable to maintain the certification which transmits a certification segment before a program segment and is used by it as a part of program authentication / authorization in alternative with all the certification used by the authentication to a user from a program variable and a user.

[0071] "File" segment processing shown in drawing 12 is performed as a result of this branch actuation. Since a file segment follows a "variable" segment typically, the check which determines whether the variable segment was already loaded (even if it was empty) is performed. If that is not right, the error will be detected and a suitable error message will be generated in 212. If the "variable" segment is already loaded, as shown in block 214, the check which determines whether the file tag of relation was already loaded to the file will be performed. If that is right, an error is detected by 216 and it is shown that the file is used as a duplex.

[0072] If the file tag is not already loaded, as shown in block 218, a file control block is assembled for a file, a tag name is set, other condition identifiers which may already have been connected with the migration program are set, and it is set about the file containing a file location.

[0073] Then, a file is read, the hash is calculated and it is kept by the segment 115 of FCB. The size of a file is kept by the segment 114 of FCB. A file does not need to be loaded in memory at this time (220).

Then, it is added to the file control block list with which the file control blocks currently created were collected by XCA, and this routine branches and returns to block 124, and processes the following segment (probably "closure").

[0074] In "closure" processing of drawing 13 , a hash is calculated from all hashes of the front about each of a front segment (230). It should be recognized that an all "segment" ingredient is hashed and read. In block 232, the check which determines whether the hash taken and calculated by 230 is in agreement with the hash added when a migration (closure segment memorizes) program was sent is performed. Error condition is set to 234 if not in agreement.

[0075] If in agreement, the check related for whether being the no with which the migration program was signed will be performed (236). notice that transmission data will not be signed at all as shown in block 238 if that is not right -- or treatment incorporating the security of any level to show and for which it asks is performed (238).

[0076] Supposing transmission is signed, a signature will be corroborated by 240 and a user will be shown the message which identifies correctly the person concerned who actually sent the migration program (and a purchase order or other formats of relation). This routine branches and returns to the block 124 of following drawing 7 .

[0077] Completion of "closure" processing of drawing 13 determines whether there is already any segment which should be processed with block 124. Then, a closure is received well and the check which determines whether it was processed or not is performed (128). If that is not right, after this routine checks failure effectiveness (130) and processes a halt by 132, it will stop activation.

[0078] If it becomes clear that the closure was well completed with the check in block 128, various steps for preparation of program execution will be performed (134). At this point, a stack is restored and a variable information table and variable control block are restored. Program control block is restored so that an activation rescue point may be included.

[0079] Then, the routine shown in drawing 14 is started and a P-code instruction is actually processed. The following problem must be examined here. That is, since it is effectively restored in the same condition as the time of program execution being transmitted from an informer's machine (as a part of traversal), it is the problem how it can distinguish whether it just returned from the machine itself been and sent in the machine which a migration program sends, or it is just going to be restored by a sink's machine.

[0080] It permits that this invention deals with this problem by many approaches. If a traversal function is realized as a built-in function, an interpreter is returned after it sends itself for a special value (for example, "0") to a program well, and when the activation is restored to a program on a sink's machine, it will return another value (for example, "1"). A migration program can test this value, in order to distinguish this situation. The option to which this distinction may be performed is because the function to extract "the number of precedence traversal" is given to a migration program (segment

98 of XCA). Before performing this traversal, a program can keep several [ a total of ] precedence-traversal-set ability using this function. If it gets to know that activation will have resumed the program by an informer's computer if it is in agreement with the value of a variable and they differ (only 1 must be large), it turns out that activation has resumed the program by a sink's computer.

[0081] When the 1st user generates a migration program, the loader routine shown in drawing 7 -13 is a very small variable, a file, or certification, or is performed without them. Therefore, the thing with the above-mentioned step will be omitted during the first processing. A loader routine is performed irrespective of whether a migration program is performed by whether it performs for the first time and the further sink.

[0082] Drawing 14 shows the actuation performed in processing of a P-code instruction. This is repeated about all P-code instructions executed. As shown in block 250, the location of the next P-code instruction is pulled out from current PCB (52), and this becomes "current" P-code actuation. In block 252, the die length of P-code actuation is determined, a "P code which is degree" location (52) is updated, and the following P-code instruction is reflected. This type of current P-code actuation is kept by (54) (this is useful although the common routine to which an interpreter has only a slight change based on exact actuation is shared.). For example, "call" actuation and "functional call" actuation are the same except a functional call expecting that a parameter will be returned. .

[0083] Then, as shown in block 254, P-code actuation shown is performed. Almost all the P-codes function includes data manipulation, a logic test, and a program flow control. Only by instantiation, such P-code actuation positions a variable. The flow control [ like ] which will reset the next P-code actuation and will be produced in branch actuation by it is changed [ pushing the variable to a stack, and ], IF/THEN/ELSE actuation is performed [ performing operation or string actuation and ] based on the stack value by which pop was carried out, DO/ITERATE/UNTIL/WHILE or other actuation are performed based on a stack value, You may also include performing SELECT/WHEN/OTHERWISE actuation based on a stack value, and performing "END" actuation and closing DO/WHEN/SELECT actuation.

[0084] Various P-code actuation is explained to a detail in relation to peculiar actuation of this invention. By guidance given here, a P-code function may be realized by the familiarity person of the store of an interpreter in an easy mode.

[0085] However, if the detail of a specific P-code function is disregarded for a while, as for a desirable design, P-code actuation permits generating logic "an interrupt" at the time of those completion.

[0086] These permit that processing of P-code processing is interrupted while some external actuation of other must be performed. Since when redundant standby or an external activity is called in a desirable design always makes the roll-out of scratch-pad memory easy, it is used by this interrupt concept.

[0087] In drawing 15 , if it returns from a P-code routine with block 256, an interpreter will determine

whether the routine sent the signal of a logic interrupt. If it has not sent, return and the next P-code actuation are dealt with to 250.

[0088] Supposing an interrupt is shown, the special check of block 258 will be performed in order that this may determine whether be the special "EXIT" demand. If that is right, all the resources that should be released by the end of these programs, such as storage, a file, a variable, and a load subroutine, will be discarded with block 260. A return value with the possibility from the P-code actuation which is probably kept by 260 is returned to the call people of this migration program with block 259.

[0089] If it assumes that this is not EXIT, as for block 261, ROLLOUT will determine whether it should be carried out or not. For example, in a certain environment, while completing that a user puts an input into scratch-pad memory, while the migration program is standing by that a time interval becomes a due date, while the redundant (or large) external program is called from migration program logic, or while the digital signature routine is performed, it is useful that a roll-out is carried out (since it includes a user input in many cases).

[0090] The routine which produces ROLLOUT with a P-code interrupt and possibility contains the following irrespective of whether they are incorporated as a built-in function or language statement (the P code of these selves is included).

[0091] SIGN -- it charges choosing a digital signature as a user from much certification in giving and doing so to any computer data, and charges giving the secret password key of the user who permits that a signature of an individual is decoded and used for a user -- I will come out.

[0092] A DISPLAY screen is constituted and outputted and a user's input supply is stood by.

[0093] Activation is interrupted until it reaches TIMEWAIT future time amount.

Selection from SELECT.FROM.DIRECTORY, for example, a user's directory, or the directory of a file is permitted.

It stands by that NOTARIZE time amount official-recognition equipment gives the digital signature of itself.

[0094] In a certain environment, ROLLOUT is unsuitable, and in these cases, the roll-out and roll-in processing in blocks 262, 264, and 268 do not exist, or will be forbidden.

[0095] As for the P-code actuation which sends the signal of an "interrupt", in any cases, the address of the function ("call-back") of at least three relation is supplied again.

[0096] -- Pre--roll-out routine. Whether being unnecessary and the becoming function are also performed in preparation of a roll-out. This may also include preparing the palm field for temporary storage and sending to ...

[0097] -- Interchange-roll-out routine. It performs, after the roll-out of as much scratch-pad memory as possible is carried out to secondary memory.

[0098] -- Post-standby routine. After an interchange-roll-out routine is completed, scratch-pad memory deals with the detail after the post-roll back restored from assistance. Typically, this includes

copying a value, as a result of being calculated by the interchange-roll-out routine which remains in temporary storage, must be loaded on activation, or must be copied in a program variable.

[0099] A pre--roll-out routine is called in block 261. This is an empty routine or may set up the parameter for for example, an interchange-roll-out routine.

[0100] In block 262, if an environment and a situation are given suitably, a roll-out function will be performed. If carried out, ROLLOUT will consist of writing VCBs and those values, FCB, certification and CCA, an activation stack, VIT and XCA, the P code itself, and all scratch-pad memory including other the blocks of any in a certain (it being (like a file)) secondary memory. This may be performed by special block (264) if it assumes that the interpreter itself remains in order to be wide opened from storage and for a sufficient residual program and data to reload an interpreter and scratch-pad memory behind.

[0101] An interchange-roll-out routine is called in step 266. Typically, it stands by that it calls another program which will require the large storage which causes other delay or was emptied by ROLLOUT or this routine stands by standing by to that a user inputs, future time amount, or other events, or an input.

[0102] In block 268, after an interchange-roll-out is completed, it reloads an interpreter and the scratch-pad memory containing an activation stack, all control block, and P codes is restored from secondary memory.

[0103] Next, in block 270, any final treatments are performed and actuation is swept away. For example, it includes that this copies the result typically returned by the interchange-roll-out routine to an activation stack or a program "a variable."

[0104] This completes an interrupt, then, control is returned to the head of a P-code handler (250), and the next P-code instruction is processed there.

[0105] Some P-code actuation which is related from this is verified. The interpreter of a desirable example deals with three CALL(s) and a function. That is, it is the routine which is dynamically positioned when it is in the routine "was incorporated" in the interpreter, the routine written in as a part of migration program and an interpreter, or the exterior of a program and a program is performed, and is called.

[0106] In drawing 17 , a built-in function is expressed quite briefly, and an interpreter only positions a specific function based on the index of a P code, searches the address of a routine (inside of an interpreter), and calls it. However, although not almost all things carry out, it is important to recognize that some built-in functions may send the signal of a P-code interrupt. In this case, a built-in function must give a required pre--roll-out, an interchange-roll-out, and a post-standby routine.

[0107] It prepares for a P-code interpreter always distinguishing CALL and a function, and returning a result to an activation stack in the case of a function. For example, a SIGN function returns the value showing the digital signature calculated on the supplied data.



[0108] In drawing 16 (A), a new PCB run level is created by 300 by the call/function to program routine. By setting the next P-code instruction (52) to the P-code entry point of a routine, new PCB is set so that activation may be started at the time of initiation of a subroutine. The instruction of the beginning of a routine will be accessed when going into block 250 again. A parameter is prepared for program routine, suitable condition conditions are set, the programme level 58 of PCB is set highly [ one ] from a calling program, and PCB is placed at the head of an activation stack as a now current PCB (82). The result of program routine is sent to call people through P-code RETURN actuation.

[0109] Program RETURN which corresponds in drawing 16 (B) P-code actuation is seen how operate. RETURN determines whether it is built with PCB of the highest level (only), this operates as EXIT in that case, and as for block 1204, block 1200 will send return RESULT as a \*\*\*\* value with which the signal of being demanded is finally returned to a P-code "EXIT" interrupt by block 261 as RESULT of delivery and the whole program ( drawing 15 ) (if it is).

[0110] Otherwise, in block 1204, a decision about whether call people used CALL or a function (for example, thing for which the field 54 is checked by call people's PCB) is made, and when it is the latter, block 1206 places return VALUE on a stack (or default value is created if RETURN does not have an operand).

[0111] In block 1208, current level is swept away and all individual (alias name "a programme level") resources are wide opened by this subroutine containing a storage area, a file, a variable, etc. A resource like the variable which is not shared with call people is not opened wide, but is available.

[0112] In block 1210, current PCB is opened wide, by it, call people's PCB becomes a current thing and activation resumes block 256 HE return and there.

[0113] An interpreter attains the specific migration program related function about these giving a digital signature and a user file to a migration program including a built-in routine, and other functions, and the need that the designer of a migration program participates in programming such a function is removed.

[0114] P-code actuation may also include the PROC actuation about the engine performance and program control block of a RETURN function which affect program control. An interpreter also performs DISPLAY actuation which uses the interactive method-of-presentation theory / language explained into this. An interpreter also performs TRAVERSE actuation and, as a result, "mailing" of the migration program is carried out like all associated data to another sink.

[0115] Drawing 18 shows in instantiation the sequence of the actuation performed in order to perform an external function or a call. Although such an external function or a call is not included in a part of interpreter or migration program, it is rather included in a part of a user's program library. The function or call which had named is positioned by 354 from either of some possible libraries.

[0116] Whether the program's having been found in 356 and the check to determine are performed. If a program is not found, as long as it is a request, the check which determines whether the program was

completed or treatment of default was performed may be performed by 358. If the decision of termination is made, an error message is generated and the post-program to which various housekeeping / clean-up actuation were carried out as mentioned above is taken out (360 362).

[0117] If the check of block 358 indicates that the measures of default should be taken, by what a special default functional value is returned for, for example (368), the measures of default are taken, and a routine will branch to the node 0 of drawing 14 , and will start activation of return and the further P-code instruction.

[0118] A parameter is constituted by the program if found as a result of the check on which the program was performed with block 356 (364). A call of an external routine includes a P-code interrupt with a possible roll-out. It should be made empty in order that the storage which permits that this has a redundant external program, or carries out a peach in what kind of environment, and saves storage in two or more users' swapping environment as an external routine is huge, therefore is used by the migration program may perform an external program to satisfaction. In this case, a P-code interrupt is sent by the signal by block 366. The PRE-ROLLOUT routine shown is copied to temporary storage from a stack (or variable) outside. An INTER-ROLLOUT routine receives a call and any returned results for an EXTERNAL routine, and a POST-WAIT routine copies the result returned (supposing the external routine was called as a function) to a stack.

[0119] As for an external routine, it is actually possible for it to be another migration program. If that is right, the already loaded image of existing of a P-code interpreter is used, and special optimization may be performed by only sending the group of a new parameter to block 120 ( drawing 7 ). In this case, probably, it will be required to avoid for special logic to be inserted in blocks 262 and 264, and to open the code of an interpreter itself conditionally.

[0120] Cautions are turned to various special built-in functions used by this example from this. These many may be performed by either as a built-in function or language statement accompanied by special P-code actuation of these selves.

[0121] Drawing 20 and 21 show the actuation performed when a migration program transmits itself to the sink who was able to set beforehand. In block 398, any program authorization information is checked first and it guarantees that traversal actuation is permitted. (Although it may not be permitted that a slight migration program moves, performing some functions only ended by the first use can consider approving.) When [ rare ] it is not permitted that a program moves, special return code is given to call people.

[0122] This example realizes "TRAVERSE" actuation as a built-in function. Furthermore, this function is prescribed to return the direct call people of a function "0" and to return "1" to the post-call people by whom the function was resumed by a sink's computer. As explained previously, this difference in return code permits that a program carries out distinction between the computers of an informer and a sink.

[0123] In order to perform this, in block 399, it turns out that a TRAVERSE function loads a value "1" beforehand on an activation stack first, and a stack remains as it is and is transmitted. This is the value which it follows and a migration program is reconfigured on a sink's computer, and will be returned when resumed. The variable datas of all relation like a "variable" information table, process control block, various stacks, and variable control block are collected in a transmission format like the format shown in drawing 2.

[0124] As shown by block 402, a migration program header is constituted and transmitted. Although a migration program is transmitted for every segment, it is the format for every field in fact, or if it is a request, some may be transmitted by the option. Preferably, it is followed for a hash on being transmitted about each segment.

[0125] Then, any authorization information from the input file received with the program and the migration program in 404 is copied to an output transmission file. A "variable" segment is transmitted including the identifier of each variable, a current value, and the condition of relation (406). this -- or any certification collected as a part which performs the pre- digital (authorization) signature in traversal is transmitted. So, when digital signature actuation is performed, in 408, the certification of all relation is always collected and transmitted to the certification section of a migration program. A signature is held as a variable within a program (namely, within variable control block). The certification in an example desirable now is treated as an ingredient which may be accessed through a built-in functional call.

[0126] In alternative, probably the program itself will be attested and it will be possible to include even the certification about the signature of transmission of the signature and the whole to permit in a certification package. However, this needs to know all certification clearly [ when a certification segment is written in ], and probably, to be again set in order will be required in order to guarantee the logic of a segment, and processing with the location optimal probably.

[0127] In this implementation, it is desirable the certification concerning the authorization signature of a program by the program authorization information in a header or a program segment and to hold the certification for the transmission signature authorization to a user from a user by the signature in a closure segment.

[0128] After certification is transmitted, all file control blocks are investigated and inspection of anythings of all the files that are probably transmitted into front traversal in 410 as a result, and the newly attached file is conducted. The check which determines whether there is still any file control block which should be investigated in block 412 is performed. The check which determines whether the schedule of the file investigated with block 414 was carried out, and it was separated is performed. If that is right, a routine will branch to 412 and neither return nor a file nor a file tag will be copied for transmission. If the schedule of the file is carried out and it is not separated, a file tag name is copied in transmission by 416.

[0129] The check which determines whether be a part of migration program by which the file of relation is carried ahead, and which enters is performed (418). If having been a part of traversal containing it is determined, all the file attributes from the traversal by which close comes will be copied to the transmission file to the exterior like the file itself (422). This input filename may be accessed through the execution control area XCA by 422, and the input location of a file relates to a file control block.

[0130] Although it is not a part of traversal containing a file, if rather attached during migration program execution, a file, a file type, and its attribute will be copied in a transmission file in 420. Then, it is determined whether a routine still has the file control block which should be investigated until it branches to block 412 and return and all file control blocks are investigated.

[0131] If the thing of all FCBs is investigated as shown in drawing 21, the digital signature from a user to the whole user will be demanded by the system program, or the check which determines whether it is needed or not will be performed by 430. Probably, the signature of such the whole will be useful in detecting modification with the transmitted information. If the whole digital signature is taken, digital signature actuation on the hash of all the transmitted ingredients will be performed (432). Digital signature actuation may be performed according to instruction of U.S. Pat. No. 5,005,200 (or idiomatic digital signature technique in which it does not have the authorization authentication attribute of relation for which it asks). As shown in block 432, the hash was beforehand taken about each part of transmission. That a hash may be taken about each of a hash in alternative attracts attention. A digital signature step may also include the user dialogue which signs.

[0132] Then, proof of justification is given to the end of transmission as a "closure" segment. Proof of justification is given by transmitting the hash reflecting a front ingredient. The signed hash should show the authentication to a user from a user in 434. Whether it being unnecessary although the justification of the last signature is proved, and the becoming certification are not already in a certification segment, either, and should be included in the CLOSURE segment. Then, transmission is closed by 436.

[0133] It is replaced with the value "0" which permits that the value "1" beforehand loaded on the activation stack when it arrived at the place whose it is a sink for the transmitted program is removed, it is returned to the present call people, and it finally distinguishes itself with block 437.

[0134] The ingredient explained to drawing 20 and 21 will operate as a P-code interruption routine in the actually desirable example including a user dialogue like asking for creating a digital signature choosing certification probably, and it opening an individual key, or operating his digital signature token equipment to a user typically. Although TRAVERSE function code carries out the trigger of the P-code interrupt and the logic to blocks 399-430 operates as a PRE-ROLLOUT routine as an example there, since the block of 432 probably needs the above-mentioned user dialogue, it may operate as an INTER-ROLLOUT routine. A block operates as a POST-WAIT routine after that (434 etc.).

[0135] A migration program may be designed so that to transmit itself to various sinks during the activation repeatedly may be wished. In such multiplex transmission, a variable may be appropriately changed in advance of each transmission. In this mode, the programs in the location which processes differ about each sink in the mode of implementation dependence.

[0136] Drawing 22 shows the sequence of the actuation for attaching a file in a migration program. An attachment file routine answers the identified file tag and the identified file name. As shown by block 440, the check which determines whether the file control block which has the same tag exists is performed. If that is right, the file before having the same tag will be deleted by 442.

[0137] Then, the check which determines whether a specific file name reflects the accessible existing file by the user is performed. At this point, a migration program may be related with the program authorization information that the operating range which a program including the capacity which accesses a file can perform is specified.

[0138] It will check in order to determine whether such program authorization information has an accessible file name. If a file name is not accessible by the user, an error code/message will be returned to a user by 446.

[0139] If the file name is accessible to a user, a file control block (FCB) will be assembled by a specific tag and a specific file name, and the file will be attached during transmission of the migration program to which it is a degree and follows by 448. A routine shows that the file is attached well after that, and is resumed.

[0140] As for drawing 23, a user system to a file shows how it is eliminated. When "elimination" function tends to be performed, in order that a security code may determine whether it was permitted that a program performs such actuation, it checks (450). If it is shown that the security code was permitted that a program eliminated a specific file (452), elimination actuation is performed, and by 454, a routine will show whether the file was eliminated well, and will branch and return. If it is not permitted in alternative that a program performs elimination actuation, a calling routine is returned with the error message which shows that a file is not eliminable (456).

[0141] Drawing 24 shows the sequence of the actuation performed in case a file is separated from a migration program. As shown in block 458, the check which determines whether a file control block exists about the identified tag about the file which should be separated is performed. If FCB does not exist, main routine is returned by 462 with the error message which shows that a file is unseparable. If it exists so that a file control block may be determined by 458, a file control block is deleted by 460, and main routine will show that the file was separated well and it will be returned.

[0142] Drawing 25 shows the sequence of the actuation performed when a file "should be exported" (i.e., when deforming into a user file). A migration program takes the specific file showing a spreadsheet program, and even after a migration program is sent to the further destination, it may change such a file into the file of the sink user who remains with a user. The file "which should be

exported" will be identified by the tag, the output file name, and the rewriting indicator that identifies whether a file may be again written in if it is a request.

[0143] It is confirmed whether a file control block exists about a specific tag first by 498. If FCB does not exist, a suitable error message code is generated and a calling routine returns to (504). If FCB does not exist with a specific tag, the check which determines whether be a part of migration program containing a file is performed by 500. If it is not a part of traversal containing the file which should be exported, the error message exporting the file which it is attached by the user, and has already existed surely in a user's file, therefore was newly attached by 502 indicates it to be not to approve will be generated. If it is a part of traversal containing a file, the check which determines whether the specific file has already existed will be performed (480). If that is right, the check which determines whether writing in a specific file again with block 482 is permitted will be performed. It includes that this check determines whether to change the existing file of specification [ a program ] (supposing there is "no exaggerated lighting"), or the paddle gap which will eliminate a specific file and will be created (if an "exaggerated lighting" is permitted) is permitted. If that is not right, block 484 will be used in order to return an access error to a program. If the check of 482 permits rewriting, a decision about whether the exaggerated lighting of the file should be carried out or a new ingredient should be added to an end of file will be made (486). An existing file will be eliminated if an exaggerated lighting is shown by 486 (488). If a permission is granted by program authorization secrecy maintenance information, a new file will be created, and preparation is made so that a store may be started at the beginning of a file (490).

[0144] Although an exaggerated lighting is not shown by 486, if new material data should be added to the end, it will be carried out as the preparation which starts adding to the existing end of file is shown in block 492. Then, data are copied to an output file from the right location in the traversal file which enters (494), show that export actuation is performed well, and are again put into main routine (496).

[0145] The sequence of the actuation to which drawing 26 is carried out when an ingredient should be signed in digital one is shown in instantiation. In implementation of a digital signature function, the check which determines whether a permission is granted by the program as digital sign actuation is first shown by block 510 is performed. It is controlled by program authorization information always supervised, when whether it is permitted that a program performs digital signature actuation performs so that it may ensure that actuation whose program cannot give authorization is not performed in relation to a program. If digital signature actuation is not permitted, the error message which refuses a digital signature function by 511 is generated.

[0146] If digital signature actuation is permitted, a user dialogue will be prepared by moving the image of the data which should be signed with a parameter (it is (like the authorization needed for the contents of data)) to temporary storage in preparation of reception by the INTER-ROLLOUT routine (shown in drawing 27 ) which will perform the user dialogue relevant to a SIGN function performing

an actual signature in block 514.

[0147] A signal is sent by the interruption routine of the after-mentioned [ a P-code routine ] in block 512.

[0148] If digital signature authorization is permitted, a display panel must be shown in order to charge to a user for signature actuation of which certification it should be used. Signature actuation is preferably performed here by citation according to this invention person's U.S. Pat. No. 5,005,200 used clearly. A user may own wide range certification, in order to perform digital signature actuation containing what was constituted along with the line of U.S. Pat. No. 5,005,200. As for an INTER-ROLLOUT routine, control is given after the roll-out of many of storage is carried out with block 509 (the signature routine itself must remain in storage of course).

[0149] If there is no suitable certification in order to sign, control will progress to block 515 and the error indicator which should be returned to sign actuation will be generated. If there is only one suitable certification to sign, it will be sent to an automatic target (513). It will be called for that a user chooses if there is many suitable certification from one (516). If a user refuses (517), a suitable error indicator will be generated and it will be sent to a program (515). Otherwise, the suitable selected certification is sent to (513).

[0150] The key of the individual of relation is positioned next (513). If block 518 determines that it is positioned on a user's token, a step (524) will be used, in order to charge a communication link to a token so that it can perform a digital signature. Otherwise, user's individual's key is positioned in the system enciphered under the secret password phrase. A user is asked for this password (520), and this is used in order to decode an individual key. If an error or an invalid password is detected, a suitable error message will be generated. Since what is guessed other than [ someone of ] a real user is forbidden, only the count to which the trial which gives a right password was restricted is permitted.

[0151] It is used in order that a password may decode an individual key in block 522, and this is used in order to sign a message according to required authorization, if it pulls. All the traces of a secret ingredient are eliminated after this actuation, and a signature and certification are returned to temporary storage (268 drawing 15 ). Control is given to the POST-WAIT routine (530) which moves a signature from temporary storage to an activation stack in (270).

[0152] In block 532, if actuation is inspected and it is carried out well, the proof hierarchy of those who sign certification will be obtained. If certification has not already appeared, it is added to the whole (held at XCA (90 etc.)) certification collection object.

[0153] Drawing 28 shows the sequence of the actuation performed in case information is displayed on a user. A migration program is related with the display layout capacity to be explained in relation to drawing 28 . The layout capacity of a migration program applies the function which related to carrying out the type set of the application for the use in a user's interactive display mode with the capacity heightened additionally conventionally.

[0154] Input field may be moved easily, and a screen may be arranged so that it may be connected with various attributes for performing an extremely flexible dialogue with a user. The actuation and the function relevant to various displays are summarized by block 540. A display presents an output based on the specific layout definition processing controlled by the display-processing part of an interpreter.

[0155] Display processing includes analysis of the condition attribute of the assembly of the field in a layout definition, and the field, and a static attribute. In a display process subroutine, it is carried out if needed the variable substitution which uses condition logic, and repeatedly. Although the variable substitution is permitted, even if this system is \*\*\*\*\* (ed) in that final output location so that the field may be directed by layout definition even if, association between the locations where it should be displayed on the screen in variable control block (VCB) to which an input variable and its field correspond is held.

[0156] A color, a font, a bold face (boldface)/italics, a format, size, an underline, a blink, a reverse video, and an attribute including non-display (for example, concealment of a password sake), a daylight display, etc. are given in each field. Furthermore, it is inserted in the suitable location for the error condition by which the possible error message was detected, and a suitable cursor location is shown.

[0157] Not only the definition of a screen output but the definition for receiving an input permits the layout language used for block 540. As shown in block 542, the field is appropriately written in the terminal of the user who permits input field by application. As stated above, after a user performs the data entry to suitable input field, the roll-out of the DS may be carried out (544), and a roll back (546) may be carried out to secondary memory.

[0158] For this reason, step 544 includes actually performing the signal of a P-code interrupt as delivery and a POST-WAIT routine which must carry out the map of the block 546 to VCB for the variable of relation, and must return block 545 for input field as an INTER-ROLLOUT routine of relation. This includes passing data through temporary storage.

[0159] Then, an input is analyzed and input data is inserted in the variable of all relation. The justification of the field is proved by 548 about all input fields. So, a check may be performed so that it may ensure being put only into a figure about much fields. The check which similarly determines whether input field have a specific attribute may be performed.

[0160] Then, the check which determines whether there was any error in either of the fields with block 550 is performed. If there is an error, an error message is generated, it will be positioned in the field which cursor mistook (552), a routine will branch to 540 after that, and return and an error message display will be generated.

[0161] If the error of a specific field cannot be clarified with the check of 550, the further check is performed and a mutual corroboration called in the field the right (for example, error condition may be separately specified [ the two adjoining fields ] for the right about the combination of the field.) is



performed by 554 in the context. Based on a mutual corroboration, a decision about whether the field includes the error of the context is made. If there is nothing, the return to call people will be performed by 558. If there is an error of the context, an error message will be generated according to block 552.

[0162] It should attract attention that the corroboration of the both sides of each field is under program control completely. Although there may be facilities to simplification of handling of various specifications, a use routine, and a common situation, the justification proof which generally has what kind of possibility is also possible. There is an inclination which may also include an interest with mutual more nearly semantic justification proof of the field, therefore needs special programming.

[0163] Drawing 29 shows the sequence of the actuation performed by the time delay routine. A time delay function may be used in order to occur with the time interval defined beforehand (wake up), performs the check which confirms whether the electronic mail into which it comes arrived, attaches itself in the mail, and it may be used in order to deal with electronic data conversion which enters by it efficiently. Therefore, it lets such a time delay device pass, and a migration program can be inspected with the time interval which was able to define the specific mail box beforehand, in order to confirm whether e-mail arrived. If e-mail has arrived, a migration program can send e-mail to the destination which should be dealt with by the further sink. In alternative, a migration program can inspect the data into which it comes (it is (like e-mail)), can be automatically traversed based on the indicator of various contents, and can cause new "moment" of itself which can process e-mail appropriately. Of course, activation and processing can be continued the first "moment" at all the arriving moments.

[0164] For example, if the information which enters is an EDI transaction by chance, a migration program breaks read-out and it, detaches information (using for example, a READ built-in function), is made into an internal variable, will determine whether it should be processed by whom and will perform suitable traversal. Once it is sent well, a letter can be disposed of, moved or recorded, a program can clear the variable, and it can resume looking for an input further.

[0165] After determining the mold of the ingredient which arrived in alternative, a call and the data into which it comes can be processed for another program. If the program of another side is a migration program by chance, the program can give required input, and it may TRAVERSE itself so that it may be suitable for handling.

[0166] This will act as an automatic router for the Iriki data [ like an EDI transaction ] whose one migration program is, and it will make it possible to pass the transaction which is not ready for dealing with itself to other migration programs.

[0167] Furthermore, as for the migration program, the signature will be immediately confirmable if EDI is signed. If the signature is effective and it will be especially performed according to U.S. Pat. No. 5,005,200, based on a program, a screen display of the authorization to the contents is carried out, and a migration program will be able to spin off an instance automatically and will be able to process the Iriki transaction.

[0168] For example, if there is raised suitable authorization, it will be possible for the buying order which enters to be automatically sent to the dispatch section in an instant, and to accept an order.

[0169] Although it arrived, if it is the article which is not signed or the article which used the signature rather simpler than the signature with authority, it will be sent to various clerks for special case processing or a more detailed inspection.

[0170] As shown in block 570, a time delay routine sets a system alarm clock to the specified time amount. Then, when the roll-out of the option of the data to secondary memory carries out scheduling of the P-code interrupt by the suitable routine, it performs (572), and the roll-in of data is performed after progress of the time amount period by which the finger-of-scorn law was carried out. The return to a calling routine occurs after that (576).

[0171] Drawing 30 shows a series of actuation for a "selection from directory" function. This directory may be the directory of a file, or a user's directory. First, a list is created by 580 from all candidate items. After that, a display is generated so that some lists [ at least ] may be displayed by 582. A user has an opportunity to choose from these shown items (583 585), and this function returns after that the identifier of the item chosen as either of the groups of a variable special as a functional result (584).

[0172] As again explained by others, actual WAIT is performed by use of a P-code interrupt function. In this case, a user minds waiting for choosing from alternative, minds a POST-WAIT routine for an input, and returns an INTER-ROLLOUT routine to a program variable.

[0173] The interpreter program of drawing 31 is [ a user ] the routine which shows how it permits performing a digital signature. As shown in block 600, the data which should be signed in digital one are assembled based on the data which can be accessed by the program. The input which the user supplied to this, the data read from the file, the data stored from front traversal, The data (for example, a user's TSO name) and time amount based on a user's environment, the data built into program itself, and the data (for example, built-in X12 data dictionary) pulled out from the built-in function are contained. Suitable information is displayed on a user (602). From it, a user determines whether he wants to sign data, as shown in block 604. If it is shown that a user wants to perform a signature, a system will call a signature function, will converse with a user further, and will complete a signature so that it may be illustrated by drawing 26 (606). After that, by 608, a digital signature is generated and it is kept as a program variable.

[0174] The flow chart following drawing 31 and it shows to some extent how while a user uses the migration programming method explained there, the powerful function which performed comparatively little actuation and was included in the above-mentioned interpreter is attained.

[0175] It illustrates how drawing 32 corroborates the information for which the user was received. As shown in block 610, the data with which what is corroborated is expected are assembled. Then, the "corroboration" function to have the authority demand in which arbitration is possible with the assembled data and the kept digital signature is called. A corroboration function uses a standard digital

signature technique, when being used in order that the conventional digital signature actuation may sign a variable as described by U.S. Pat. No. 5,005,200 or, and it may be attained. Then, it is determined whether the signature was corroborated based on processing of the block circuit 12 (614). Program execution will be continued if that is right. If that is not right, it will be shown that the data was also unfairly changed by 616, or error condition had the programming error of some classes. It is specified that return code enables a program whether a signature is invalid, whether it supported authorization capacity, and to distinguish whether the authorization was checked if that was right.

[0176] It is [ how drawing 33 collects the files to which a migration program should be transmitted, and ] \*\*\*\*\*. First, a program determines the file which should be transmitted by displaying the list of files on a user by 620. A check may be performed in order to determine whether it is necessary to carry out a user dialogue in order to determine a file (622). If it is yes, a user will be urged to determine the file which should be transmitted by 624. If it is not necessary to perform a user dialogue in order to determine a file, the whole file content will be attached to the group of the data which should be transmitted by 626. Actuation is attained using the added function which was stated to drawing 22 , and this drawing includes building a file control block, as explained above.

[0177] Drawing 34 illustrates the migration program actuation performed in case data are read from the specified file. The file containing the data which should be read first is determined (630). Then, by 632, it is read from the file as which data were specified, and is kept as a program variable. As for drawing 35 , a migration program illustrates how a file can be updated or created from a program variable. As shown in block 640, the user file by which data should be written in into it is determined first. Then, the function which writes a program variable in a user file by 642 is called.

[0178] Even if explained to cases clearly [ no ], if it is the program function which may lead to a data loss, modification, damage, or discovery, anything to be under rule of security control must be understood. Such control is given with a programme level, or is connected to the Iriki program, and is performed by being together put in what is probably already imposed by the beam user, and a certain mode defined beforehand.

[0179] therefore -- or [ for example, / that, as for a migration program a program reads an authority attachment \*\*\*\*\* user's data file such in an above-mentioned case ] -- or it can only write in.

[0180] Security constraint exists to the function of the following classes at least. Data are displayed on a user.

[0181] An input is charged from a user. A digital signature is performed.

[0182] Data are read from a user file. A user file is created.

[0183] A user file is eliminated. Data are written in a user file.

[0184] A user file name is changed. A user file is added.

[0185] The added file is taken out to a user file. Digital authentication equipment is called.

[0186] The E-mail which carries out Iriki is received. The contents of the E-mail are read.

[0187] The mail which carries out Iriki is moved or recorded. The mail which carries out Iriki is eliminated.

[0188] The E-mail which goes to the exterior is created, or data transmission various type is performed.

[0189] It is combined with equipment various type, a device, and services (FAX, a printer, an office device, robot equipment, manufacturing installation, etc.).

[0190] Program traversal is performed. An external program is called.

[0191] other migration programs -- access, updating, activity-izing, elimination, modification, and a call -- or it adds.

[0192] Illustrating how drawing 36 may be designed so that it may be sent to the addressee from whom the migration program was divided and a large number differed, drawing 37 shows how the program divided before may be merged.

[0193] A migration program may need to divide a migration program, in order to gain survey data from the addressee from whom a large number differed with reference to drawing 36 first, or in order [ or it collects data, ] to distribute data to the management from whom much organizations differed. First, a migration program performs various housekeeping actuation, in order to prepare division by 650. Then, a variable is set up according to the survey performed by specific application demand, for example, a specific user, by 652. A destination user is determined by 654, a traverse function is called according to drawing 20 and drawing 21, and other suitable data of the arbitration which suited each addressee with the image of a program and the program variable are transmitted. The transmitted variable may change from an instance 1 (656) to an instance 2 (658), an instance 3 (660), and Instance N (662).

[0194] In order to determine whether there are many destinations rather than it should transmit to it, finally a check is performed (664). If that is right, a routine will branch to 652 and will transmit to return and the further destination. If the further destination cannot be found, a final transfer will be performed in the same mode as having been explained to 654 in the top 666, will bring about final "instance" by 668, and will become completion of division actuation after that.

[0195] In other examples, a master program can only progress to other processings of some kind. Supposing it is probably performing in a batch environment like an input distributor, and if all inputs are used up now, it will be delayed until some of other things arrive (about [ that many users spun off ]).

[0196] With reference to merge actuation of drawing 37, a migration program transmits itself to a user from a user, and it has the intelligence by which data are merged further until merge actuation is completed. First, a migration program reaches a merge destination and is performed (680). A check is performed in order to determine whether be the master "an instance" determined by setting up the variable with which this was defined beforehand. If it is determined that this is not a master instance in 682, a slave instance will be identified by 684. By (685), supposing it confirms how [ that was called with the special "DEBRIEF" parameter ] it is (this is mere agreement used by this program in order to

determine when the slave is called by the master) and meets (687), all suitable information will return and leave a slave program to a master instance. If this is not a DEBRIEF call, in order to determine [ that is, ] whether a master instance is available and whether it has already arrived, a check will be performed by 686. If the master instance is available, the call to a master instance will be performed by 696 by the use of a call shown in drawing 18 . After a master instance is called, a routine branches and returns to block 680. If a master is not available, the message that the master control to the continuation has not arrived yet by 688 will be emitted.

[0197] A master instance arrives, and a check is performed, in order it to be determined by block 682 for that this is a master instance, assuming that it was called and to determine whether any of other slave instances arrived by 692. If that is right, a slave instance will be called with the parameter defined beforehand, and will start collection of data by 694 (probably called "debriefing (debriefing)"). Data are collected from an instance by the entry point A, and it is returned to a master by 706, and is written in a collection file. An instance to the extent that it was called after that is eliminated by 708, a routine branches to 692 and return and the further information on whether other instances arrived in that case are collected.

[0198] If other further instances have not arrived, the file is checked in order to find whether all the instances of all arrived (698). If they have arrived, as determined by 700, data may be read into the variable of a migration program from collection. The file completed at the moment may be processed depending on the size a collection file is expected to be, and the property of processing, and itself may be traversed to the following destination, or the result may be protected carefully, and it may be more desirable for a master program a simple message and to make it a \*\* EDI transaction probably and to only transmit the raw data.

[0199] It may be appropriate for a program to ATTACH a file at itself in other cases, and to transmit it to another process in large quantities. The file is eliminated by 704 and aggregate data is transmitted to the following destination. If all instances have not arrived yet, a message like "waiting for a format to arrive" is published (702), and it is made to leave a routine temporarily.

[0200] Drawing 38 shows the approach of an alternative to merge the migration program information divided before. As shown by block 710, a migration program reaches a merge destination and is performed. The collected data are specially written in a file by 712. A check is performed in order to determine whether all other instances arrived, as shown in 714. If that is right, the collected data will be processed by 716, you will traverse a program to the following destination by 718, and it will be made to leave a routine. If it has not arrived as all other instances were determined by 714, a message like "waiting until more formats arrive" is displayed (720), a current instance is eliminated by 722, and it is made to leave a routine.

[0201] It is the flow chart which shows how drawing 39 was designed in order that a migration program might hold an electronic data exchange (EDI) generation function. Drawing 39 shows more

concretely how the specific "X12" standard description may be used. X12 criterion has a related data dictionary and a related segment dictionary. X12 segment dictionary may be used in order to specify all segments required in order to specify a buying order. It is specified that each segment is data of the piece investigated by the dictionary after that. Since the amount of an item is specified and there is an approach by which many differed, many change of data is permitted by X12.

[0202] This system inserts X12 data dictionary in the interpreter which may be called a built-in function. As shown by block 720, the call to X12 subroutine is performed at first by specifying before a segment name and an item "XX, YY, WW .." A program can give X12 data code to the common option which has typical popularity in the environment of an organization, and it is possible to, build the short list of options as a result for anticipated use. The examples of such an item are the number of items, the number of components, and an amount in the buying order context. This call turns into a call to a built-in data dictionary.

[0203] A check is performed in order to determine whether the short list is empty (it is shown by 724 like). If that is right, it will be used in order to call built-in function X12SEGLIST to which a segment name carries out location of the segment dictionary table of all related data options by 736. Then, it is used in order that a X12DATANAME built-in function may extend the explanation data each related by 738 using a data dictionary, and a perfect long list is displayed by 740.

[0204] If it is shown that there is a list with the short check of 724, X12DATANAME data dictionary will be used in order to carry out location of each extended description of the option of a short list. A list short after that at 728 is displayed. A check is performed in order to determine whether desire a perfect long list as the user indicated to be by 730 after that. If the answer is yes, block 736 will be performed as mentioned above. if -- no -- if come out and it is -- a short list -- or selection of the user from either of the long lists is accepted (732).

[0205] A check is performed by block 734 in order to determine whether all data are collected. If that is right, we will assemble publish X12 transaction completed by 742, and will make the routine leave. this invention mails the whole migration program to the issue actuation referred to in relation to 742 -- in addition, the capacity which mails X12 data of a specific group is considered carefully. If it is collected as data are shown [ no ] by the check of 734, more data items are searched and routine activation is repeated.

[0206] Drawing 40 relates to the application of the migration program at the time of receiving an electronic data exchange transaction. For example, possibly the specific user received the buying order which the migration program generated. First, the received EDI transaction is read by 750. Probably, as explained by timer delay using drawing 29 , the migration program which produces the copy of itself as an input arrives. The syntax of encoded EDI is analyzed by the program variable by 752 from it. It is moved to a records depot from it, and received EDI saves what was received for possible audit. This segment is processed through the segment dictionary combined by 756. The segment regulation

relevant to X12 may be tightened up, and it may relate to it not having the kind of the specific field of a certain kind of data by 758. Location of the data dictionary relevant to each segment is carried out by 760 to each data item. To the statement as shown by 762 which is DESC=X12DATANAME (SEGCODE, DATA ITEM), this statement will bring a result of the call to a data dictionary, and will obtain description with the semantics of a data item. The description with the searched semantics is set to a display variable, and becomes the display of the buying order of the result, for example, a buying order format. All data items are processed by branching and returning to block 762, and all segments are processed by branching and returning to 756.

[0207] This desirable example permits access to a digital authentication device by giving digital authentication, authentication equipment which is explained by this invention person's U.S. Pat. No. 5,001,752 (it uses here by citation), or the built-in function which can access other equipments similarly again.

[0208] By permitting that a migration program accesses such a device, a migration program can move data to a platform, and it is accessed easily, and digital authentication uses a built-in function there, in order to make it such. It signs and this makes possible authentication for the time stamp for other reasons of some kind for the important traffic which enters. Since such authentication is under severe control of a program, the criteria of arbitration may be used when it is based on a user demand, if it will probably be automatic.

[0209] As explained above too, this device is taking into consideration association to FAX left outside, and as a result, in addition to being changed into EDI or being printed, the electronic format is possible also for carrying out FAX to a final addressee, and is made.

[0210] Moreover, even when the migration program has emitted the EDI transaction as shown tacitly although not stated clearly, it may still be activity-ized later. An example is a migration program which functions as an electronic demand first and generates a buying order after sufficient approval signature. It is again activity-ized, when the invoice and bill corresponding to a depository finally arrive itself after that delivery and there (in mode of electronic or others), and it may function as an approach for making it in agreement with loading and the bill which received the order. Which item is received and it can incorporate the logic which pursues which is still pending. For the capacity to orient itself flexibly, it may spread in the location where many differed. As long as loading and a receipt are treated, it is also possible to combine a migration program with a bar code reader, and to prove delivery and the ingredient received without human being's data input.

[0211] It assumes that a desirable example may be combined with various equipments with which a migration program includes an office device, other equipments, and a device.

[0212] Moreover, the given traversal of arbitration may be sent to various addressees at coincidence. The following lists repeat and summarize many of above-mentioned functions which can be performed by the desirable example (and some additional functions are identified). This list is only

instantiation and what is shown without the place which leaves much other application with which this invention may be applied advantageously is not meant. Data are displayed on a user using layout language (for example, similar to TeX or SCRIPT).

[0213] An input is charged from a user using layout mold language (similar to TeX or SCRIPT).

[0214] The digital signature for the data calculated under the program control is performed.

[0215] A digital signature is corroborated based on the data calculated under the program control.

[0216] A joint signature is processed including probably sending the proposal pulled out from a signer's certification.

[0217] Data are read from a user file. A user file is created.

[0218] A user file is eliminated. Data are written in a user file.

[0219] A user file name is changed. The E-mail which carries out Iriki is received.

[0220] The contents of the E-mail are read. The mail which carries out Iriki is moved or recorded.

[0221] The mail which carries out Iriki is eliminated. The E-mail which goes outside is created.

[0222] It is controlled and it combines with the FAX server which goes outside. It is controlled and it combines with a printer.

[0223] A graphical image is created. It is controlled and it combines with the device which can transmit and receive an audio signal.

[0224] The equipment containing an office device, computer machines (a tape, disk, etc.), robot equipment, a manufacturing installation, etc. various type is accessed.

[0225] Multiplex traversal divides the instance of a migration program into some instances.

[0226] It is possible to recombine with a single format the data contained in some migration programs, without even carrying out that the same program is probably shown.

[0227] Other instances of a migration program are eliminated. An external program is called.

[0228] Other migration programs are called as a subroutine. Other migration programs are activity-ized as a function performed independently.

[0229] Data are extracted from a dormant (un-performing) migration program. The information about others and a migration (un-performing) program is determined without being accompanied by the needs of performing it, such as an identifier of a program, and other conditions.

[0230] Information is extracted from the certification relevant to a digital signature. This information is used in order to help to send directly, if a joint signature demand is included.

[0231] The copy of a migration program is built as a data variable within other programs, or a migration program is ATTACH(ed) to other things as a file.

[0232] One migration program ("carrier") is used in order to convey the high version of other things to various destinations, and the program segment of an existing instance is replaced with another thing which is the newest version from that of the program. One approach for carrying out like this is making it added to the edge of a migration program at which a newer program segment is existing. It



will recognize that the program segment to which enhancement of existing interpreter/loader follows a closure segment reflected the proposed program revision. It will be begun whether to use a new program instead of the program which arrived the digital signature relevant to the proposed program which was revised as a part of standard traversal if they were performing a real proof and suitable authority, after [ what kind of ] transmission is also usually performed.

[0233] A user file is added. The added file is taken out to a user file.

[0234] The file added before is removed. Digital authentication equipment is accessed.

[0235] Program traversal is performed. User data are transmitted (except traversal) and it is made for the transmission not to contain migration program itself as a result (for example, a message is only sent to other destinations).

[0236] The built-in function for simplifying use of EDI (for example, X12 or EDIFACT), creation, a display, construction, and reception is used, and common information and a common device are supplied conveniently, without being accompanied by the need of supplying these functions of a migration program. This includes the built-in function which accesses a data element dictionary, a segment dictionary, a segment regulation, and the transaction set itself.

[0237] Although this invention has been explained in relation to what is considered to be the present most practical example, it must be understood that this invention is not what is restricted to the indicated example, and it is meant that it covers various corrections included in the pneumonia of a claim shown above and within the limits on the contrary and the array of equivalence.

.....  
[Translation done.]

\* NOTICES \*

**JPO and INPIT are not responsible for any damages caused by the use of this translation.**

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

.....  
DESCRIPTION OF DRAWINGS  
.....

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram of the communication system according to the instantiation-example of the invention in this application.

[Drawing 2] It is drawing showing a migration program and the instantiation-structure of the related component.

[Drawing 3] It is drawing showing instantiation-execution control field DS.

[Drawing 4] A migration program is drawing showing the DS of the file control block (FCB) used in a file in case a file is removed from installation or itself to itself.

[Drawing 5] It is drawing showing process control block used in the case of migration program execution.

[Drawing 6] It is drawing showing the variable control block data structure (VCB) used in order to control a variable.

[Drawing 7] It is drawing of an instantiation-migration program loader.

[Drawing 8] It is drawing showing signs that a header is loaded.

[Drawing 9] It is drawing showing signs that the "program" segment of a migration program is loaded.

[Drawing 10] It is drawing showing signs that the "variable" segment of a migration program is loaded.

[Drawing 11] It is drawing showing signs that "certification" segment of a migration program is loaded.

[Drawing 12] It is drawing showing signs that the "file" segment of a migration program is loaded.

[Drawing 13] It is drawing showing signs that the "closure" segment of a migration program is loaded.

[Drawing 14] It is drawing showing the actuation performed in the case of processing of a P-code instruction.

[Drawing 15] It is drawing showing the processing performed after P-code actuation is performed.

[Drawing 16] (A) And (B) is drawing showing the processing for dealing with the function or call by program convention.

[Drawing 17] It is drawing showing the sequence of the actuation for dealing with a built-in function.

[Drawing 18] It is drawing showing the sequence of the actuation performed in order to perform an external function or a call.

[Drawing 19] It is drawing showing the sequence of the actuation performed in order to perform an external function or a call.

[Drawing 20] It is drawing showing the actuation performed in case a migration program mails itself to the sink who was able to set beforehand.

[Drawing 21] It is drawing showing the actuation performed in case a migration program mails itself to the sink who was able to set beforehand.

[Drawing 22] It is drawing showing the sequence of the actuation for attaching a file to a migration program.

[Drawing 23] A file is drawing showing signs that it may be eliminated from a user's system.

[Drawing 24] It is drawing showing the sequence of the actuation performed in case a file is removed from a migration program.

[Drawing 25] It is drawing showing the sequence of the actuation performed when a file is changed in a user's file.

[Drawing 26] It is drawing showing the sequence of the actuation performed when data are signed in digital one.

[Drawing 27] It is drawing showing the sequence of the actuation performed by the "INTER-ROLLOUT" function.

[Drawing 28] It is drawing showing the sequence of the actuation performed in case information is displayed to a user.

[Drawing 29] It is drawing showing the sequence of the actuation performed by "time delay" routine.

[Drawing 30] It is drawing showing the sequence of actuation for a "selection from list of names" function.

[Drawing 31] It is drawing showing the routine which shows signs that it enables a user to perform a digital signature by the interpreter program.

[Drawing 32] It is drawing showing signs that a user corroborates the received information.

[Drawing 33] It is drawing showing signs that the files which a migration program should be delivered are collected.

[Drawing 34] It is drawing showing actuation of the migration program performed in case data are read from the specified file.

[Drawing 35] A migration program is drawing showing signs that a file can be updated or made from a program variable.

[Drawing 36] It is drawing showing signs that it is designed so that a migration program may be divided, and a program can be sent to the sink from whom many differ.

[Drawing 37] It is drawing showing signs that the program divided in advance may be merged.

[Drawing 38] It is drawing showing the alternative-like policy which merges the migration program information divided in advance.

[Drawing 39] It is the flow chart which shows the mode designed so that a migration program may accept an electronic document interchange generating function.

[Drawing 40] It is drawing showing use of the migration program at the time of receiving an electronic data exchange transaction.

[Description of Notations]

2 -- Processor W / main memory

4 -- A keyboard/CRT

6 -- Modem

12 -- Communication channel

.....

[Translation done.]